ScreenQA: Large-Scale Question-Answer Pairs Over Mobile App Screenshots

Yu-Chung Hsiao[*†] **Fedir Zubach**[*] **Gilles Baechler** **Srinivas Sunkara**
Victor Cărbune **Jason Lin** **Maria Wang** **Yun Zhu** **Jindong Chen**[‡]

Google DeepMind Cisco Systems[†]

yohsiao@cisco.com, {fedir, baechler, srinivasksun,
vcarbune, jasonjlin, mariawang, yunzhu, jdchen}@google.com

Abstract

We introduce ScreenQA, a novel benchmarking dataset designed to advance screen content understanding through question answering. The existing screen datasets are focused either on low-level structural and component understanding, or on a much higher-level composite task such as navigation and task completion for autonomous agents. ScreenQA attempts to bridge this gap. By annotating 86k question-answer pairs over the RICO dataset, we aim to benchmark the screen reading comprehension capacity, thereby laying the foundation for vision-based automation over screenshots. Our annotations encompass full answers, short answer phrases, and corresponding UI contents with bounding boxes, enabling four subtasks to address various application scenarios. We evaluate the dataset's efficacy using both open-weight and proprietary models in zero-shot, fine-tuned, and transfer learning settings. We further demonstrate positive transfer to web applications, highlighting its potential beyond mobile applications.

1 Introduction

Recent advancements in machine learning, especially Visual Large Language Models (VLMs) or Multimodal LLMs (MLLMs), have catalyzed numerous applications centered on mobile screens. These applications range from personal assistants enabling hands-free or eyes-free interaction to code generation from UI design mock-ups, adaptive device interfaces, automated ad creation, and mobile app testing. All of these require reliable screen content understanding as the foundation to ensure quality applications.

Mobile app screenshots have been analyzed using machine learning from multiple perspectives. These include pixel-level understanding, such as layout structural analyses, UI issue detection and correction (Li et al., 2022), UI element semantics like icon recognition, button action prediction (Sunkara et al., 2022), to even higher-level functional analyses such as accessibility support (Li et al., 2020b), screen description (Wang et al., 2021), and screen type classification (Deka et al., 2017). However, the specific aspect of understanding screen contents from pixels remains comparatively understudied.

Screen contents encompass diverse information, from restaurant ratings to system settings and chat messages. This capacity is crucial because: 1) screens primarily function to present information, and 2) autonomous agents and task completion systems require precise screen content understanding to achieve reasonable success rates for multi-step processes.

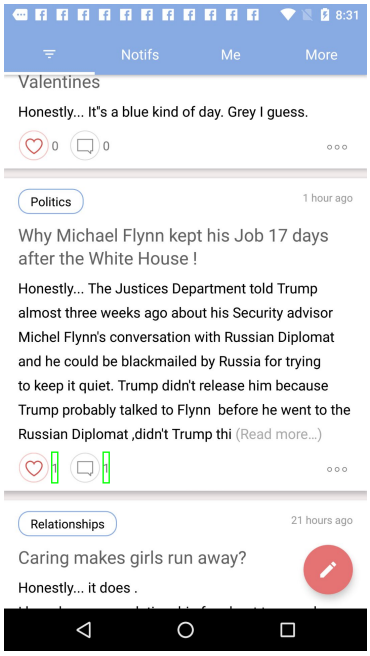
In this work, we propose using pixels as the sole representation of UI screens, to eliminate dependency on unreliable structural representations such as view hierarchies (VHs) (Zang et al., 2021) and maximize applicability to various visual understanding tasks. To this end, we annotated the RICO dataset (Deka et al., 2017) with 85,984 question-answer pairs (Section 4) and defined four tasks with evaluation metrics (Section 3), creating *ScreenQA*[1].

RICO remains the preferred screenshot dataset due to its 1) extensive collection of 66k screenshots, 2) diverse app distribution spanning 9.3k apps across 27 categories, 3) numerous benchmarking tasks built upon RICO that enable cross-referencing and comparison[2] (Wang et al., 2021; Li et al., 2020b, 2022; Ahmed et al., 2023; Lu et al., 2024), and 4) widespread use as a training dataset in recent document/screen understanding and VLM research (Cheng et al., 2024; Liu et al., 2024b; Xie et al., 2023). Despite the availability of

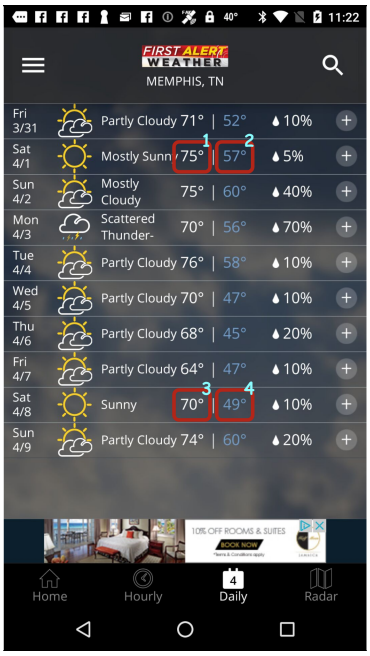
[*] Co-first authors with equal contributions. [†] This work was done while the author was at Google. [‡] Corresponding author.

[1] ScreenQA dataset is released at https://github.com/google-research-datasets/screen_qa [2] We deliberately chose not to combine RICO with other datasets for this reason.

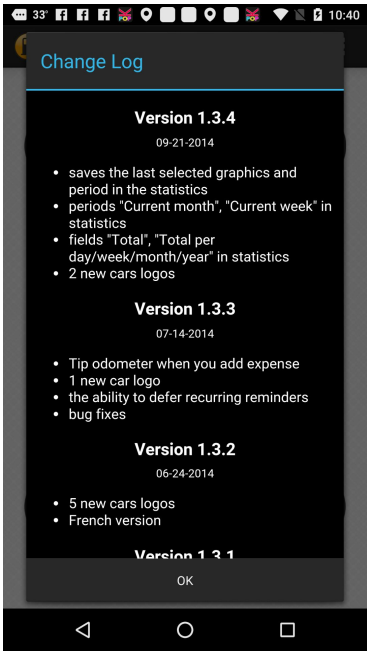
Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 9427–9452



(a) Question: "How many likes and comments are there for the post *Why Michael Flynn ...*?"



(b) Question with Ambiguity: "What's the temperature on Saturday?"



(c) No answer to the question: "What is the date of version 1.3.1?"

Figure 1: ScreenQA examples. (a) Four corresponding tasks (Section 3): 1) Short answer (SQA-S): "1,1". 2) Long answer (SQA-L): "There is 1 like and 1 comment". 3) UI Content (SQA-UIC): ["1", "1"]. 4) UI Content w/ Bounding Boxes (SQA-UIC-BB): also with bounding box coordinates (in green). (b) Both high and low temperatures in two Saturdays are relevant answers. (c) Lacking the content to answer the question.

newer datasets, RICO's comprehensive nature and established presence in the field make it an optimal choice for this work. See Table 1 for comparison.

This work offers the following contributions:

1. We create and release ScreenQA, the first large-scale question answering dataset and benchmark for mobile screens.
2. ScreenQA provides rich annotations: short answers, full-sentence long answers, and supporting UI element contents with their bounding boxes (Section 4).
3. We propose four tasks with evaluation metrics to satisfy various application needs in extraction, text generation, and grounding (Section 3).
4. We establish zero-shot, fine-tuning and cross-domain transfer learning baselines for both closed and open-weight models quantifying the effectiveness of ScreenQA (Section 6).

2 Related Work

Closest to ScreenQA is work that tackles multimodal understanding and question answering, as discussed below.

Dataset	Size	Unit	Apps	Categories
RICO (Deka et al., 2017)	66k	screenshot	9.3k	27
LabelDroid (Chen et al., 2020)	13k	screenshot	7.6k	25
MoTIF (Burns et al., 2022)	4.7k	trace	0.1k	n/a
AitW (Rawles et al., 2024)	715k	trace	0.4k	n/a

Table 1: The RICO dataset remains the most suitable mobile app screenshot collection for this work due to its diversity, large scale, and established presence in the field. Although AitW is larger in scale, it lacks app and screenshot diversity due to its sampling method.

2.1 Multimodality

We discuss the research on text-heavy images[3] and provide an overview in Table 2.

Screen UI-based understanding Unlike natural images, UIs are designed to be informative and actionable. Examples of work dealing with informativeness include 1) UI element identification, such as icon detection (Deka et al., 2017), widget captioning (Li et al., 2020b; Chen et al., 2020), and 2) referring expression to UI elements in classification (Wu et al., 2023), representation learning (Bai et al., 2021) and generation (Hong et al., 2023).

[3] Natural image VQA and video VQA are omitted due to the space constraints.

Dataset	# Images (k)	# Examples (k)	UI?	Task Type
ScreenQA (this work)	35 *	86	✓	Question Answering (QA)
TextVQA (Singh et al., 2019)	28	45	✗	QA
DocVQA (Mathew et al., 2021)	12	50	✗	QA
InfographicVQA (Mathew et al., 2022)	5.5	30	✗	QA
ChartQA (Masry et al., 2022)	22	33	✗	QA
WebSRC (Chen et al., 2021a)	6.5	440	✓	QA on Webpage Segments
ComplexQA (Baechler et al., 2024)	10	12	✓	QA on Counting, Arithmetic
VisualWebBench - WebQA (Liu et al., 2024a)	0.3	0.3	✓	QA on Webpages – Eval Only
LabelDroid (Chen et al., 2020)	13	19	✓	Text Generation
Screen2Words (Wang et al., 2021)	22	112	✓	Summarization
ScreenAnnotation (Baechler et al., 2024)	22	22	✓	Object Detection
MoTIF (Burns et al., 2022)	62	4.7	✓	Navigation
AitW (Rawles et al., 2024)	5,690 †	715	✓	Navigation

* A subset of the 66k images from RICO (Deka et al., 2017) as described in Section 4
† Limited app and screenshot diversity as described in Table 1

Table 2: Comparison of ScreenQA with Related Datasets. ScreenQA is the largest QA dataset for mobile screenshots, using entire screenshots rather than cropped answer regions (as in WebSRC) to better represent real-world applications, and including unanswerable questions and bounding boxes. ComplexQA complements ScreenQA with its focus on counting, arithmetic, and comparisons. Not an exhaustive comparison due to space constraints.

Actionability is related to task completion and autonomous agents. MoTIF (Burns et al., 2022), VisualWebArena (Koh et al., 2024), and Android-in-the-Wild (Rawles et al., 2024) provide interactive app environments for evaluating visually grounded screen agents. This work is focused on the information retrieval aspect.

Visual Document Understanding Visual Document Understanding concerns understanding scanned or photographed documents. DocVQA (Mathew et al., 2021) uses an extractive QA format for span/segment extraction. TextVQA (Singh et al., 2019) and several other domain-specific datasets (Mishra et al., 2019; Gurari et al., 2018; Huang et al., 2019; Abdallah et al., 2024; Jaume et al., 2019; Harley et al., 2015; Lewis et al., 2006) relate the 2D arrangement of texts to semantic meanings. Beyond text alone, infographics understanding (Mathew et al., 2022; Masry et al., 2022; Kahou et al., 2017; Kafle et al., 2018; Chaudhry et al., 2020) focus on charting with text around.

2.2 Question Answering

We focus on closed-domain question answering grounded in specific contexts. Answer formats include span (Rajpurkar et al., 2016), entity (Talmor and Berant, 2018), multiple choice (Mihaylov et al., 2018), and generation (Xiong et al., 2019). Capacities range from reading comprehension (Yang et al., 2015), multi-hop reasoning (Yang et al., 2018), (Chen et al., 2021b), logical reasoning (Yu et al., 2020), and commonsense reasoning (Talmor et al., 2019).

3 Problem Setting: Tasks and Metrics

As each example in the validation and test sets may contain alternative ground truths from multiple annotators, we compute the evaluation metric by: 1) calculate the maximum metric value across all annotator-provided ground truths for each example, and 2) average these maximum values across the entire dataset:

$$\text{avg}(metric) = \frac{1}{N}\sum_{i=1}^{N} \max_{j}[metric(A_i, A^g_{i,j})],$$

where N is the number of questions, A_i is the predicted answer for i-th question, and $A^g_{i,j}$ is the j-th ground truth for i-th question.

ScreenQA (or SQA in short) presents four tasks encompassing diverse application scenarios. Examples are illustrated in Figure 1.

ScreenQA Short answer (SQA-S) Given a screenshot and a question, output a short (concise) answer to this question using the information presented on the screen. If the screenshot doesn't contain the answer, output "<no answer>".

Similar to SQuAD (Rajpurkar et al., 2016), we propose to use *Exact Match* (EM) and *F1-Score* for accommodating acceptable permutations and rephrasing of the same content. We also apply SQuAD pre-processing to normalize answers before computing the metrics. This task is the core capability enabled by the dataset.

ScreenQA Long answer (SQA-L) This task is identical to SQA-S, with the distinction that it requires a long, full-sentence answer instead of short answer phrases. This task facilitates the generation of coherent responses suitable for direct human interaction, particularly in the context of virtual assistant applications.

As the task resembles summarization of pertinent information, we propose to use *ROUGE-{1,2,L}* (Lin, 2004) as the evaluation metric.

ScreenQA UI Content (SQA-UIC) Given a screenshot and a question, output a list of UI elements that contain the answer, where each element is represented by its text representation. If the screenshot doesn't contain the answer, output an empty list.

Section 4 will further details the UI elements corresponding to questions, their listed order, and their contents represented in text. Except for icons with predefined textual descriptions, most content resembles OCR output: text within UI elements (Qin et al., 2019). However, unlike OCR, the output should be evaluated as a list rather than a continuous sequence of symbols or words.

We use *Exact Match* and *F1-score* as metrics. Commonly, text in screenshots is directly extractable. Therefore, we perform UI-element-wise matching without additional pre-processing.

ScreenQA UI Content with Bounding Boxes (SQA-UIC-BB) Given a screenshot and a question, output a list of UI elements that contain the answer, where each element is represented by its bounding box *and* text representation.

This task extends the previous SQA-UIC, additionally facilitating answer highlighting and action automation over the screen. The detection of bounding boxes, especially in screen contents, is rarely available in existing datasets and challenges model capabilities.

We recommend evaluating the bounding box detection quality using *F1-Score*, where two bounding boxes match if their Intersection over Union (IoU) (Rezatofighi et al., 2019) score is higher than 0.1. *Exact match* and *F1-score* are evaluated in a similar way, but restricting the list of matches to only those where text representation also matches. This threshold is justified due to the annotation methods of ground truth bounding boxes: selection from view hierarchies (accommodating UI elements with substantial no-content area) or manually drawing (tightly fitting text) (Section 4.3). The simultaneous use of these approaches by different annotators typically results in low IoU values.

Stage	Step	# SS	# Q
	RICO Original	66k	–
Prefiltering Section 4.1	Non-English apps†	11k	–
	Out-of-sync VHs†	13k	–
	For Question Anno.	51k	–
Question Anno. Section 4.2	1st pass of Q anno.	35k	46k
	2nd pass of Q anno.		36k
	Lack of content	15k	–
	For Answer Anno.	35k	82k
Not Ans. Anno. Section 4.4	3rd pass of Q anno.	5k	5k
	For Data Splitting	35k	86k
Data Splitting Section 5.1	Train	28,378	68,951
	Validation	3,485	8,614
	Test	3,489	8,419
	Total	35,352	85,984

† Not mutually exclusive.

Table 3: Counts of distinct screenshot (SS) and questions (Q) for annotation stages and data splitting.

4 Data Annotation

We performed a five-stage process to annotate ScreenQA: 1) Prefiltering (Section 4.1), 2) Question annotation (Section 4.2), 3) Answer annotation (Section 4.3), 4) Not-answerable question annotation (Section 4.4), and 5) Short answer generation (Section 4.5). See data statistics in Table 3, data annotation details and UI tools in Appendix A, and data examples in Appendix B.

4.1 Prefiltering

Prefiltering involves one round of human rating to remove screenshots with the following conditions:

1. Screenshots from non-English apps.
2. Screenshots with unsynchronized view hierarchies (VHs).

Removing these data helps avoid language-related obstacles and potential view hierarchy issues (Zang et al., 2021), which could otherwise introduce noise into subsequent annotation stages. We employed 27 annotators to perform this stage. Occlusion and ghosting during screen transitions are deemed acceptable if the UI elements in the main content area remain clean and accurate, resulting in different numbers from (Li et al., 2020a). Examples of such VH symptoms are provided in Appendix A.1.

4.2 Question Annotation

We employed the same 27 annotators to formulate questions based on provided screenshots, simulating real-world app usage queries. Questions were restricted to directly observable information on the screen, deliberately excluding those questions requiring reasoning, computation, or advertisement-related content. This is because recent multimodal reasoning benchmarks (Yue et al., 2024; Han et al., 2023) unveil that the multimodal LLM reasoning capacities are heavily influenced by their underlying LLM performance (Section 6 of (Wang et al., 2024)), which can be more efficiently achieved through inference time scaling (Brown et al., 2024; Snell et al., 2024; Wu et al., 2024; Kumar et al., 2024), followed by pretraining data and model parameter scaling (Zhang et al., 2024; Owen, 2024). By focusing our scope on screen content understanding, we aim to investigate multimodality independent of testing other types of intelligence.

Question annotation was a two-pass process. First, an annotator composed up to five questions per screenshot. Then, a second annotator, with access to the initial questions, added up to three more, resulting in a maximum of eight questions per screenshot. This two-pass approach promoted a diverse and comprehensive set of questions, while reducing redundancies. To encourage diverse questioning styles and a varied set of questions, each annotator was required to complete their own initial set of questions before they could see questions created by others. Screenshots lacking interesting content (e.g., login pages, ads) may be skipped.

4.3 Answer Annotation

Given a screenshot and a question, the annotators are tasked to

1. Correct grammatical errors in the question.
2. Answer the question by: a) Annotating bounding boxes that constitute the answer. b) Ranking bounding boxes by relevance or reading order.
3. Provide a full-sentence long answer.

The annotator who composed a specific question is excluded from answering it to prevent potential bias. Our UI tool offers two bounding box annotation methods: direct selection of UI elements from view hierarchy(VH) leaf nodes, or manual drawing when VH structure is inadequate for selection.

We further consider two exceptions: 1) Mark "invalid question" if incomprehensible. 2) Mark "not answerable from the screenshot" if the screenshot does not contain the answer. The "invalid question" answer annotations are first removed for each question. Then questions without any remaining answer annotations are excluded from the ScreenQA dataset, as they are deemed invalid. To enhance evaluation quality, three answer were annotated for validation and test sets, while only one for training set (details in Section 5.1).

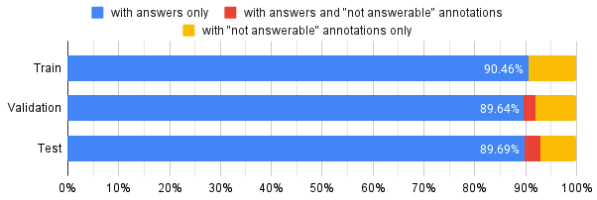


Figure 2: Distribution of question answerability. Each split contains approximately 10% questions that have at least one "not answerable" answer annotation, to test models' ability to refrain from answering unanswerable questions.

4.4 Not-Answerable Question Annotations

A third pass of question annotation was implemented to increase "not-answerable questions" to approximately 10% (Figure 2). This tests models' ability to refrain from answering unanswerable questions, similar to (Rajpurkar et al., 2018). Approximately 5k screenshots without existing "not answerable" questions were randomly sampled. We instructed annotators to compose one question per screenshot that was related to the screen content but unanswerable. See an example in Figure 1c.

4.5 Short Answer Generation

As UI design of apps may fragment a short, semantic answer into multiple bounding boxes (e.g., "Oct. 15, 2024" fragmented into "Oct.", "15", and "2024" in a date selector), we perform a post-processing step to make the annotations into a coherent, human readable short answer suitable for common question answering purposes. This also partly accommodate answer variants that are semantically identical, such as numerical representations (digits/words), date/time formats, and optional descriptors/units (e.g., "3 reviews" vs. "3"). We employed PaLM 2 (Anil et al., 2023) to generate answer variants via few-shot prompting with inputs including question, UI element descriptions, and full-sentence answer, with a subset verified by authors. See Appendix A.6 for the prompt details.

5 Dataset Analysis

5.1 Dataset Splitting

The ScreenQA dataset comprises 85,984 questions from 35,352 unique screenshots, split at the screenshot level into training, validation, and test sets in an approximate 80-10-10 ratio (see Table 3).

5.2 Question Analysis

Among the 86k collected questions, 47.5k are unique in term of SQuAD EM. Common questions often recur across screenshots (e.g., "Which option is selected?" or "What is the email address?"). Screenshots with more information typically generate more questions. The histogram of questions per screenshot follows a mild exponential decay (Figure 3a). We further categorize questions using regular expressions, providing a preliminary overview in Table 4. It is worth noting that the category distribution is implicitly influenced by RICO's crawling process. For example, the crawling process commonly includes a step at the login page, resulting in a higher percentage of questions about app names, email addresses, and login permissions.

5.3 Answer Analysis

We analyze the answer annotations in two aspects: 1) How often multiple bounding boxes are required for answers, indicating task complexity, and 2) How often VH is sufficient for bounding box annotation, indicating the reliability of VHs.

Figure 3b illustrates the histogram of bounding boxes per answer. About 84% of answers contain a single bounding box, with 51% utilizing VH leaf nodes and 49% using manually drawn boxes. When all answers are considered, 51% rely solely on VH leaf nodes, 48% on manual boxes, and 0.8% on a combination. Combinations typically occur for multi-part answers involving diverse UI elements, scattered components (e.g., dates), or measurements with units. Despite VHs being commonly used in prior screen tasks (Burns et al., 2022; Li et al., 2020b), the near-equal preference for manual bounding boxes reflect VH limitations: VHs cannot capture UI elements in WebViews and Canvas, and are inconsistent for certain screen designs. This justifies our decision to use screenshot as input in ScreenQA.

6 Experiments and Baselines

We conducted three sets of experiments: zero-shot, fine-tuning, and transfer learning, each of which is explained below.

6.1 Zero-Shot Experiments

We evaluated Fuyu-8B (Bavishi et al., 2023), Gemini 1.5 Flash, Gemini 1.5 Pro (Gemini Team Google, 2023), and GPT-4o (OpenAI et al., 2024) on SQA-S in a zero-shot setting. Results are summarized in Table 5. We prompted each model using an instruction followed by a question. For the Fuyu-8b model, we used the instruction that model's authors recommended in their examples. For GPT-4o, we iterated on the prompt design using a set of 10 examples from the ScreenQA validation split and then reused the prompt for Gemini 1.5 models. See Appendix D.1 for the detailed prompts. Despite the prompt being originally designed for GPT-4o, Gemini 1.5 Pro outperforms GPT-4o in this particular setup.

6.2 Fine-Tuning Experiments

We fine-tuned three series of models spanning open-source, domain specific, and general purpose proprietary models on the ScreenQA tasks as described below.

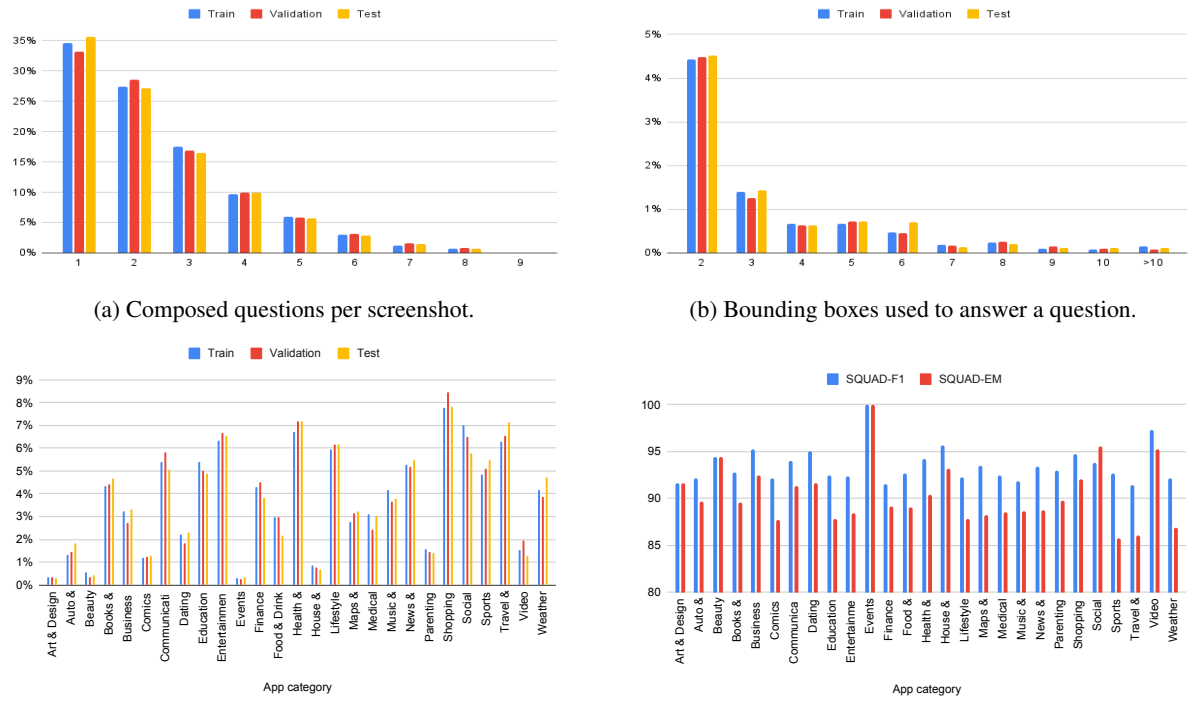
PaliGemma 3B We used the pre-trained PaliGemma 3B model checkpoints (Beyer et al., 2024) with three input resolutions, 224×224, 448×448 and 896×896, and fine-tuned for 10 epochs with a learning rate 1.0×10^{-5} using the Adam optimizer and a cosine decay schedule. Both vision and language backbones were trained during fine-tuning.

ScreenAI 670M and 5B This VLM specializes in UI and infographics understanding (Baechler et al., 2024). We used a dynamic 812×812 input resolution and fine-tuned this model until convergence, using a learning rate of 1.0×10^{-3}. Only the language backbone was trained during fine-tuning.

Gemini 1.5 Flash We also fine-tuned Gemini 1.5 Flash (Gemini Team Google, 2023) model[4] on the downstream tasks for ~ 7 epochs with a learning rate of 1.0×10^{-4} for SQA-S and SQA-L and 1.0×10^{-6} for SQA-UIC and SQA-UIC-BB. Only the language backbone was trained during fine-tuning.

Note that we did not test PaliGemma and ScreenAI models' zero-shot performance as they

[4] https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini-use-supervised-tuning



(a) Composed questions per screenshot.

(b) Bounding boxes used to answer a question.

(c) Questions per app category.

(d) Fine-tuned PaliGemma 896 performance on SQA-S.

Figure 3: Histograms for various data and model performances. (b) Approximately 91% of questions are either not answerable or can be answered by a single bounding box, hence, omitted to emphasize the long tail distribution. (d) The model exhibits consistent SQuAD-F1 performance across app categories, except for Events and Videos, which present higher scores due to lower support for those categories.

are not instruction-tuned, therefore, not suitable for evaluation in a zero-shot setting.

We use the metrics introduced in Section 3 to measure the performance of models along various dimensions: i) Extracting relevant information for answering a question in the SQA-S and SQA-UIC tasks, ii) Ability to provide fluent answers in the SQA-L task, and iii) Identifying relevant UI elements through their bounding box coordinates in the SQA-UIC-BB task.

We present the results in Table 6. We observe a slightly higher performance for the ScreenAI 5B model compared to PaliGemma 3B, and attribute it to its larger model capacity and specialized pre-training mixture that includes a rich variety of UI elements. The ScreenAI model also performs on par with the Gemini 1.5 Flash model in a fine-tuned setting. We also notice a larger difference between the ScreenAI model and other approaches in the SQA-UIC-BB tasks involving bounding box prediction. PaliGemma performance is however very competitive, and by fine-tuning both language and vision backbones we enable better use of the entire model capacity.

In Appendix E we present a few examples of the different errors made by the models and a categorization of these errors.

6.3 Cross-Domain and Transfer Learning

We conducted cross-domain learning (CDL) and transfer learning (TL) by fine-tuning PaliGemma models (Table 7). For CDL, we fine-tuned the models using either SQA-S or DocVQA (Mathew et al., 2021) and evaluated on the other. These experiments aim to characterize the differences between ScreenQA and other VQA datasets from related domains like document images by investigating the performance difference by training on related datasets vs. training on the corresponding training split. We observe that the use of in-domain training data results in significant improvements for both SQA-S and DocVQA datasets. However, we do note that for the SQA-S task, using training data from the related domain of document images results in performance gain over the zero-shot setting with a larger Fuyu-8B model (Table 5), showing that they are related but distinct domains.

For transfer learning, summarized in Table 7, we used VisualWebBench-WebQA (Liu et al., 2024a), a recently released QA task on web screenshots,

Category	%	Examples	
UI selection & config	18.1	Which option is selected?	What is the selected ringtone?
Quantity number	11.7	How many unread messages?	How many pictures are there in Western Europe?
App name	10.4	What is the name of the application?	What is the app name?
Date time	9.4	When was “Heal the Living” released?	When is happy hour?
Price	3.4	How much is the gift bonus in 3rd place?	What is the price?
Name of item	3.3	What is the name of the drug?	What is the name of chef?
User name	2.8	What is the name of the user?	What is the username on telegram?
Duration	2.5	What is the duration of video?	How long is the song?
Enum. of avail. options	2.5	Which social media options are given there?	What are the options available for logging in?
Address and direction	2.4	What is the current location?	What is the service zip code?
Email address	2.4	What is an email address?	What is customer service email?
Person’s name	2.1	Who sang the song?	What is the last name?
Others	12.8	What’s the average speed?	What is the user’s middle initial
Subtotal	83.8	What is the spending limit?	Which team has 41 points?

Table 4: Top question category distribution ($\geq 2.0\%$) and examples (See Appendix C for remaining categories).

	SQuAD-EM	SQuAD-F1
Fuyu-8B	39.5	47.3
Gemini 1.5 Flash	80.6	86.4
Gemini 1.5 Pro	**81.4**	**87.2**
GPT-4o	77.8	86.6

Table 5: Zero-shot public models evaluation on SQA-S. Bold is best performance.

as the target evaluation dataset. We fine-tuned the models using DocVQA alone and using both DocVQA and ScreenQA, demonstrating positive transfer when incorporating the ScreenQA dataset.

Additionally, we tested the hypothesis that training models to predict bounding boxes would enhance their performance on tasks not explicitly involving bounding boxes. For example, fine-tuning models on both SQA-S and SQA-UIC-BB tasks and evaluating performance on SQA-S task. However, in our experiments with smaller models such transfer benefits did not materialize. One possible explanation could be that this is due to increased complexity caused by introducing an additional task. While this may seem like a limitation, it also reflects the standalone challenges each task presents, reaffirming the value of our dataset.

6.4 Importance of Multimodality and Fine-Tuning

We used Gemini 1.5 Flash (Gemini Team Google, 2023) model and SQA-S and SQA-L tasks to demonstrate the complexity of the screen data for screen understanding. Experiments in Table 8 show that using text representation of the screen information (OCR) as model input produces significantly worse results than using multimodal (image & text) setup. Noticeable difference in performance between zero-shot and fine-tuned setups suggests that screen understanding domain presents its own challenges on top of generic image understanding.

7 Conclusion

We introduced ScreenQA, a rich dataset that enables training and evaluating models on question-answering tasks on screen content. We described the annotation process, statistics of the collected dataset, which contains 85,984 question-answer pairs. In addition to answers, our dataset contains extensive annotations of the UI elements, enabling the ability to train or probe models for their holistic understanding of the screen, a necessary capability for high-level reasoning and automation using UI interfaces. Compared to other vision-language tasks, such as document understanding or visual-question answering, the four constructed tasks on the ScreenQA dataset pose unique challenges: rich in text, diverse in mobile applications, blended with icons and symbols. The tasks not only evaluate content quality, but also UI element identification quality. Furthermore, we conducted a diverse set of experiments, including zero-shot, fine-tuned, and transfer learning, on a series of open-weight and proprietary models to assess the capacity of our ScreenQA dataset. We encourage the community to tackle screen content understanding challenges present in our benchmark, fostering new technologies and user experiences.

8 Limitations

We acknowledge limitations of our work.

Language Released data is only in English and further work would be necessary to create a multilingual variant of our dataset. This is further amplified by the need of screenshots with phones configured in the different locales.

Model	SQA-S		SQA-L			SQA-UIC		SQA-UIC-BB		
	EM	F1	R-1	R-2	R-L	EM	F1	BBOX-F1	EM	F1
ScreenAI 670M	51.2	60.6	77.3	68.4	76.7	47.8	49.4	62.7	41.1	42.6
PaliGemma 3B 224	77.5	83.9	88.2	81.5	87.4	74.8	76.7	84.9	67.5	69.6
PaliGemma 3B 448	88.3	92.2	91.1	85.5	90.3	86.0	87.7	89.4	79.1	81.6
PaliGemma 3B 896	89.4	93.2	90.9	85.3	90.1	86.1	87.8	88.8	78.8	81.2
ScreenAI 5B	**90.7**	94.6	**92.6**	**87.4**	**91.9**	87.0	88.7	**94.2**	**84.0**	**85.7**
Gemini 1.5 Flash	90.5	**94.9**	92.4	86.2	91.7	**88.2**	**89.7**	92.4	83.9	**85.7**

Table 6: Model performance for ScreenQA tasks after fine-tuning. Bold is best performance.

Experiment | Task	**CDL** | SQA-S				**CDL** | DocVQA		**TL** | VisualWebBench-WebQA	
Fine-Tuned on	SQA-S		DocVQA		SQA-S	DocVQA	DocVQA	SQA-S + DocVQA
Model \ Metrics	EM	F1	EM	F1	ANLS	ANLS	F1	F1
PaliGemma 3B 224	77.5	83.9	52.6	60.5	27.5	43.7	19.31	**21.51**
PaliGemma 3B 448	88.3	92.2	66.2	72.9	55.1	78.0	48.33	**49.11**
PaliGemma 3B 896	89.4	93.2	63.6	70.4	62.0	84.8	57.07	**58.69**

Table 7: Cross-domain learning (CDL) and transfer learning (TL) experiments via PaliGemma model fine-tuning. For CDL, models were fine-tuned on either SQA-S or DocVQA and evaluated on the other. In-domain and cross-domain model scores exceed 60 for the 896×869 model and differ by 20∼25 points across the board (underlined). TL experiments revealed positive transfer when additionally using SQA-S for the WebQA evaluation (bold).

Evaluation setup	SQA-S		SQA-L			SQA-UIC		SQA-UIC-BB		
	EM	F1	R-1	R-2	R-L	EM	F1	BBOX-F1	EM	F1
Zero-Shot, Text-Only	64.4	72.5	78.2	67.8	75.6	38.6	41.2	30.1	49.7	28.5
Zero-Shot	80.6	86.4	83.8	70.8	79.3	62.4	66.8	26.2	33.3	24.0
Fine-Tuned	90.5	94.9	92.4	86.2	91.7	88.2	89.7	85.7	92.4	83.9

Table 8: Gemini 1.5 Flash performance against ScreenQA for various evaluation setups. Note that for SQA-UIC-BB, the Zero-Shot, Text-Only case (the 1st row) outperforms its multimodal counterpart (the 2nd row) because the bounding box coordinates are given as text input: the model just needs to select the right UI elements with the given coordinates, which makes bounding box prediction very precise when they are correct, hence, an easier task.

Rich Layouts Our data and annotation process focuses on rather static content, facilitating us to focus on information extraction from canonical layouts. There are a number of challenges our dataset does not capture, specifically rich layouts coming from gaming or other highly interacting applications that may be present on a user's device. Screenshots that contain natural images and videos are also missing.

Reasoning The type of challenges our dataset focuses on are a lot more related to information look-up and rewriting. We therefore capture challenges grounded in UI content and composition understanding, rather than arithmetic or other forms of complex compositional reasoning challenges. See Section 4.2 for details.

Multi-Image An example in our dataset always consists of a single screenshot. Several challenges that come with multi-image understanding such as a trace of screenshots are therefore not covered by our tasks in the dataset. Similarly, no UI animations are included or scrolling actions.

Platform ScreenQA contains screenshots only from the Android ecosystem and the phone form factor. As all of the composed questions are focused on the main content area without involving any gestures and actions, we do not anticipate major performance differences across ecosystems. In addition, our experiments show positive transfer for VisaulWebBench-WebQA when additionally using ScreenQA (Table 7), indicating the efficacy of ScreenQA for other form factors.

9 Ethical Considerations

Information Retrieval Only ScreenQA focuses on information retrieval of screen contents, with the primary intent of improving screen content understanding. ScreenQA dataset does not involve any

materials related to decision-making for or against users, nor does it execute actions on behalf of users, avoiding potential harm.

Privacy The technologies fostered by ScreenQA, when used on a mobile device, require access to information at a level comparable to that of a standard accessibility application. Furthermore, the promising results from fine-tuned PaliGemma models (Table 6) suggest that the technologies enabled by ScreenQA can be effectively hosted on mobile devices, significantly limiting potential privacy concerns.

App Category Fairness ScreenQA contains questions across 27 categories of apps (Figure 3a), which is the most diverse set at the time of writing (Table 1). Also, our fine-tuned PaliGemma 896 model exhibits consistent SQuAD-F1 performance across these app categories (Figure 3d), demonstrating that ScreenQA has a fair support across app purposes.

References

Abdelrahman Abdallah, Mahmoud Abdalla, Mahmoud SalahEldin Kasem, Mohamed Mahmoud, Ibrahim Abdelhalim, Mohamed Elkasaby, Yasser ElBendary, and Adam Jatowt. 2024. CORU: Comprehensive Post-OCR Parsing and Receipt Understanding Dataset. *Preprint*, arXiv:2406.04493.

Ali Ahmed, Alaa Zaki, Enas Elgeldawi, Mohamed Abdallah, and Moheb Girgis. 2023. MASC: A Dataset for the Development and Classification of Mobile Applications Screens.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. PaLM 2 technical report. *arXiv preprint arXiv:2305.10403*.

Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Cărbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. 2024. ScreenAI: A vision-language model for UI and infographics understanding. *Preprint*, arXiv:2402.04615.

Chongyang Bai, Xiaoxue Zang, Ying Xu, Srinivas Sunkara, Abhinav Rastogi, Jindong Chen, and Blaise Aguera y Arcas. 2021. UIBert: Learning generic multimodal representations for UI understanding. *Preprint*, arXiv:2107.13731.

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. 2023. Introducing our multimodal models.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. 2024. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling. *Preprint*, arXiv:2407.21787.

Andrea Burns, Deniz Arsan, Sanjna Agrawal, Ranjitha Kumar, Kate Saenko, and Bryan A. Plummer. 2022. A dataset for interactive vision language navigation with unknown command feasibility. In *European Conference on Computer Vision (ECCV)*.

Ritwick Chaudhry, Sumit Shekhar, Utkarsh Gupta, Pranav Maneriker, Prann Bansal, and Ajay Joshi. 2020. LEAF-QA: Locate, Encode & Attend for Figure Question Answering. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3501–3510.

Jieshan Chen, Chunyang Chen, Zhenchang Xing, Xiwei Xu, Liming Zhut, Guoqiang Li, and Jinshui Wang. 2020. Unblind your apps: Predicting natural-language labels for mobile GUI components by deep learning. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 322–334.

Xingyu Chen, Zihan Zhao, Lu Chen, Danyang Zhang, Jiabao Ji, Ao Luo, Yuxuan Xiong, and Kai Yu. 2021a. WebSRC: A Dataset for Web-Based Structural Reading Comprehension. *Preprint*, arXiv:2101.09465.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021b. FinQA: A Dataset of Numerical Reasoning over Financial Data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. SeeClick: Harnessing GUI Grounding for Advanced Visual GUI Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9313–9332, Bangkok, Thailand. Association for Computational Linguistics.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A Mobile App Dataset for Building Data-Driven Design Applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pages 845–854, New York, NY, USA. Association for Computing Machinery.

Gemini Team Google. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018. VizWiz Grand Challenge: Answering Visual Questions from Blind People. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3608–3617, Salt Lake City, UT, USA. IEEE.

Xiaotian Han, Quanzeng You, Yongfei Liu, Wentao Chen, Huangjie Zheng, Khalil Mrini, Xudong Lin, Yiqi Wang, Bohan Zhai, Jianbo Yuan, Heng Wang, and Hongxia Yang. 2023. InfiMM-Eval: Complex Open-Ended Reasoning Evaluation For Multi-Modal Large Language Models. *Preprint*, arXiv:2311.11567.

Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 991–995.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. 2023. CogAgent: A visual language model for GUI agents. *arXiv preprint arXiv:2312.08914*.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shjian Lu, and C. V. Jawahar. 2019. ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. *Preprint*, arXiv:1905.13538.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. DVQA: Understanding Data Visualizations via Question Answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5648–5656, Salt Lake City, UT. IEEE.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. FigureQA: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2024. Training Language Models to Self-Correct via Reinforcement Learning. *Preprint*, arXiv:2409.12917.

D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 2006. Building a test collection for complex document information processing. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 665–666, New York, NY, USA. Association for Computing Machinery.

Gang Li, Gilles Baechler, Manuel Tragut, and Yang Li. 2022. Learning to Denoise Raw Mobile UI Layouts for Improving Datasets at Scale. In *CHI Conference on Human Factors in Computing Systems*, pages 1–13, New Orleans LA USA. ACM.

Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. 2020a. Mapping Natural Language Instructions to Mobile UI Action Sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8198–8210, Online. Association for Computational Linguistics.

Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. 2020b. Widget Captioning: Generating Natural Language Description for Mobile User Interface Elements. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5495–5510, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. 2024a. VisualWebBench: How Far Have Multimodal LLMs Evolved in Web Page Understanding and Grounding? *Preprint*, arXiv:2404.05955.

Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. 2024b. TextMonkey: An OCR-Free Large Multimodal Model for Understanding Document. *Preprint*, arXiv:2403.04473.

Yuwen Lu, Yuewen Yang, Qinyi Zhao, Chengzhi Zhang, and Toby Jia-Jun Li. 2024. AI Assistance for

UX: A Literature Review Through Human-Centered AI. *Preprint*, arXiv:2402.06089.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. DocVQA: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2200–2209.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. OCR-VQA: Visual question answering by reading text in images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 947–952. IEEE.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

David Owen. 2024. How predictable is language model benchmark performance? *Preprint*, arXiv:2401.04757.

Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, and Ying Xiao. 2019. Towards unconstrained end-to-end text spotting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2024. Android in the wild: A large-scale dataset for android device control. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 59708–59728, Red Hook, NY, USA. Curran Associates Inc.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. *Preprint*, arXiv:1902.09630.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8317–8326.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. *Preprint*, arXiv:2408.03314.

Srinivas K. Sunkara, Maria Wang, Lijuan Liu, Gilles Baechler, Yu-Chung Hsiao, Jindong Chen, Abhanshu Sharma, and James W. Stout. 2022. Towards Better Semantic Understanding of Mobile Interfaces. In *Proceedings of the 30th International Conference on Computational Linguistics*, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. The Web as a Knowledge-Base for Answering Complex Questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. 2021. Screen2Words: Automatic Mobile UI Summarization with Multimodal Learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, pages 498–510, New York, NY, USA. Association for Computing Machinery.

Yiqi Wang, Wentao Chen, Xiaotian Han, Xudong Lin, Haiteng Zhao, Yongfei Liu, Bohan Zhai, Jianbo Yuan, Quanzeng You, and Hongxia Yang. 2024. Exploring the Reasoning Abilities of Multimodal Large Language Models (MLLMs): A Comprehensive Survey on Emerging Trends in Multimodal Reasoning. *Preprint*, arXiv:2401.06805.

Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. 2023. Webui: A dataset for enhancing visual ui understanding with web semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. An Empirical Analysis of Compute-Optimal Inference for Problem-Solving with Language Models. *Preprint*, arXiv:2408.00724.

Jinheng Xie, Kai Ye, Yudong Li, Yuexiang Li, Kevin Qinghong Lin, Yefeng Zheng, Linlin Shen, and Mike Zheng Shou. 2023. Learning Visual Prior via Generative Pre-Training. *Advances in Neural Information Processing Systems*, 36:70562–70580.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. TWEETQA: A Social Media Focused Question Answering Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5020–5031, Florence, Italy. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for

Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. ReClor: A Reading Comprehension Dataset Requiring Logical Reasoning. *Preprint*, arXiv:2002.04326.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI. *Preprint*, arXiv:2311.16502.

Xiaoxue Zang, Ying Xu, and Jindong Chen. 2021. Multimodal icon annotation for mobile applications. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*, pages 1–11.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. 2024. When Scaling Meets LLM Finetuning: The Effect of Data, Model and Finetuning Method. *Preprint*, arXiv:2402.17193.

A Data Annotation Details

A.1 VH Out-of-Sync Rules

We asked annotators to mark screenshots which are 1) from non-English apps and 2) not synchronized with VHs, as described in Section 4.1. The out-of-sync symptoms are oftentimes harder to distinguished than expected. UI elements may be occluded (Figure 4a) and "ghosting" VHs appear in addition to the in-sync VHs because of UI animation effects such as menus sliding/popping in and out (Figure 4b). These two cases are deemed to be acceptable in-sync scenarios as a user or an annotator can still select the correct bounding boxes from VH. However, the example in Figure 4c is an out-of-sync case, in which the text "No Alarms. Set an alarm and start your ..." is not supported by a VH bounding box. All the other bounding boxes that appear on the screen are all irrelevant to the current main content, hence, determined as an out-of-sync case. We trained our annotators to distinguish these cases and only mark the third case to remove.

A.2 Question Annotation UI

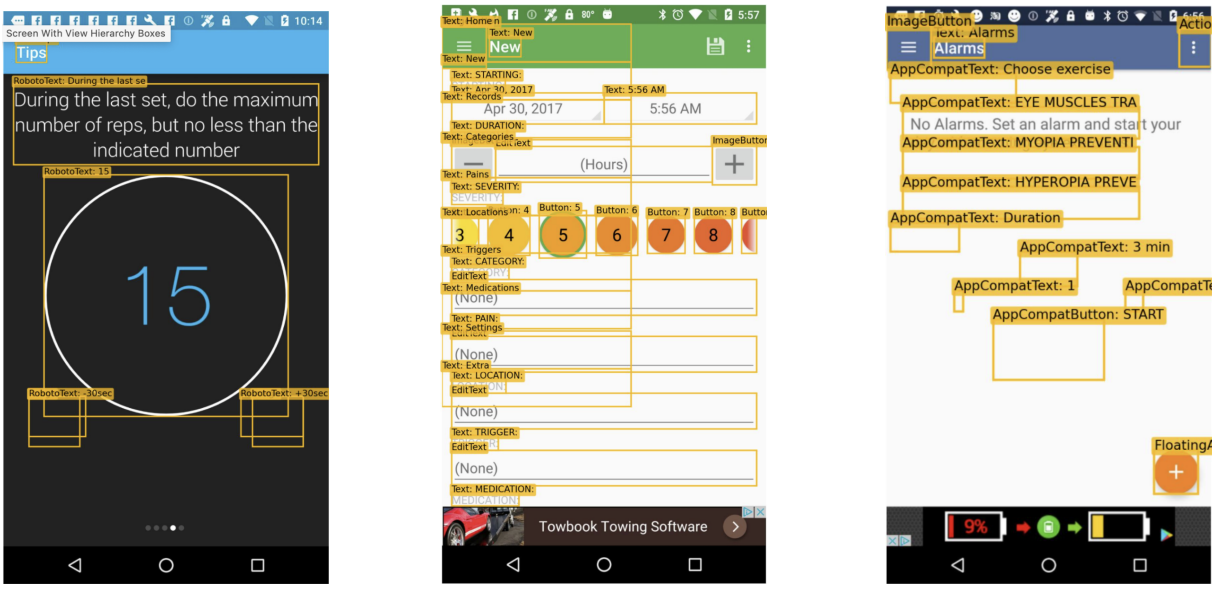
The question annotation interface is shown in Figure 5a. Question annotation was performed in a sequential manner by multiple annotators. An annotator can see all previous questions to diversify question framing and avoid duplication. We also used the same sequential process to provide more feedback and training to the annotators for quality improvement.

A.3 Answer Annotation UI

The answer annotation interface is shown in Figure 5b. Answer annotators were tasked to determine if the question is valid and if the question is answerable from the screen context. If both are positive, the annotators need to answer the questions by 1) selecting or drawing the bounding boxes of UI elements, 2) filling the text for each selected/drawn bounding box on the right, 3) ranking them appropriately, 4) providing the full-sentence answer to the question. The annotators were also tasked to review and make necessary corrections if the question has grammatical errors or typos.

A.4 Annotation quality control

We leveraged a rating platform for collecting human annotations to ensure that raters go through



(a) VH with occluded elements.

(b) Ghosting VH from menu.

(c) Out-of-sync VH for main content.

Figure 4: View hierarchies (VHs) are overlaid on the screenshots with class names and the first few characters printed to assist annotators to determine whether the VHs for the main contents are in sync.

a training phase, and proceed to collecting annotations only after reaching 90% quality bar. Furthermore, output quality was monitored throughout the annotation process, consistently reaching more than 95%.

A.5 Annotation post-processing

Aside from removing answer annotations marked as "invalid question" and questions with no remaining answer annotation, post-processing of the records in the train split included applying question modification, if there was one provided during answer annotation.

The same could not be easily done for validation and test splits, as those contained 3 answer annotations per question, and corresponding question modifications could vary. The inter-annotator agreement was therefore estimated. The 2 answer annotations were considered to be in agreement if any of the following conditions apply:

- The questions (after modifications if some were provided) are the same.
- The full-sentence answers are the same.
- The lists of bounding boxes that constitute the answer contain the same elements, ignoring permutations.
- Both answer annotations are in agreement with the same other answer annotation.

Here two bounding boxes correspond to the same element if they correspond to the same VH node, or if they intersect and have the same textual content. As a result, 97.9% of the questions in validation and test splits had full agreement of the answer annotations, 1.9% had partial agreement (2 out of 3), and 0.2% had no agreement.

We then removed the disagreeing answer annotations from questions with partial agreement, and questions with no agreement. And applied question modification with the biggest consensus (2-3 out of 3), or chose the latest if all modifications are different.

A.6 Short Answer Generation Prompts

We describe below the prompts used in PaLM 2 (Anil et al., 2023) to generate short answers for ScreenQA, as described in Section 4.5. ScreenQA dataset annotations (question, list of text of UI elements, and full-sentence answer) were used as input. To improve the prompt quality, we used an identical prompt template with two sets of few-shot examples for answers with 1) one or 2) more than one UI elements involved, each set of which is used and inserted into the template for the example with answers that utilize the corresponding number of UI elements. The prompt template is give below:

```
List various ways to rephrase the answer. The answer
 should be as short as possible, without extra words
 from the question. Use all provided elements in
each answer. Provide the output in square brackets.

{examples}

Now is your turn.
Question: {question}
Answer elements: {list of text of UI elements}
Full answer: {full-sentence answer}
Rephrases:
```

An example of single-UI-element answer is as below:

```
Here is another example:
Question: 'What is the gender?'
Answer elements: ['Male']
Full answer: 'The gender is male.'
Rephrases: ['male']
```

An example of multiple-UI-element answer is as below:

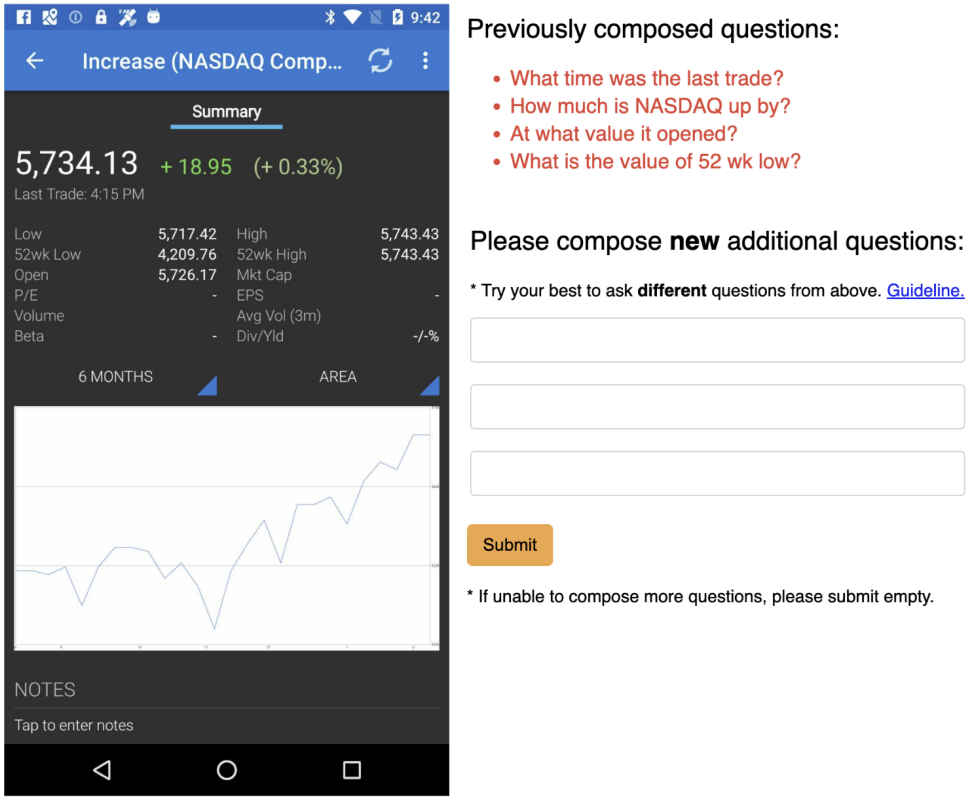
```
Here is another example:
Question: 'What is the name?'
Answer elements: ['Jon', 'Brown']
Full answer: 'The name is Jon Brown.'
Rephrases: ['Jon Brown']
```

B Data Examples

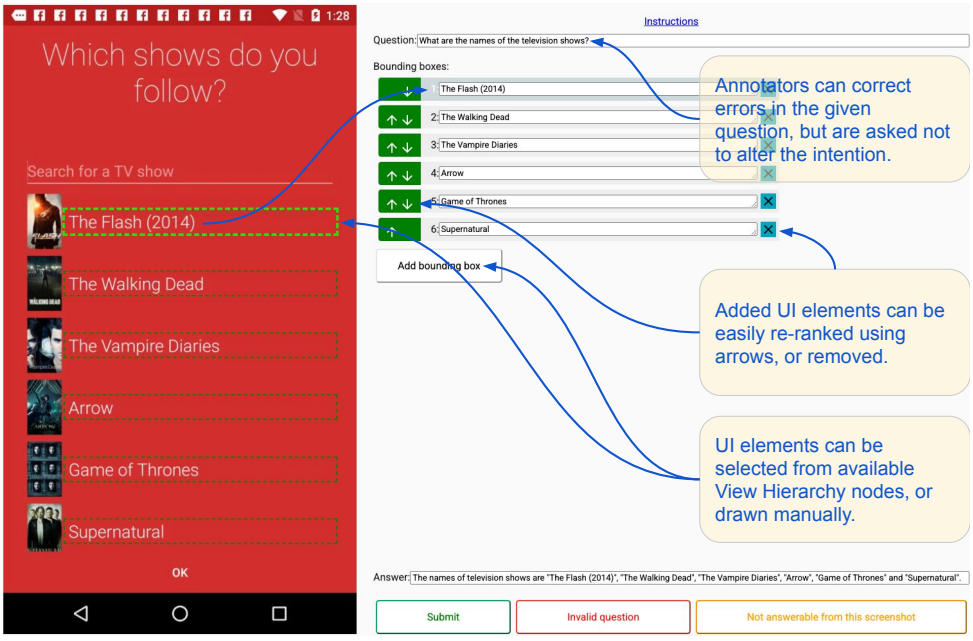
Table 9 presents a few examples from the ScreenQA dataset. Each example contains

- A screenshot
- A question
- Multiple annotations of lists of UI elements relevant to the question, each annotation of which is completed by an annotator
- Multiple annotations of long full-sentence answers corresponding to the UI element annotations
- Multiple short answers, generated by the procedure outlined in Section 4.5

Note that the bounding boxes of selected UI elements are highlighted on the screenshot for the illustrated purposes of the corresponding annotated bounding boxes, but they are not present in the corresponding image from RICO (Deka et al., 2017) or during our data annotation process outlined in Section 4.

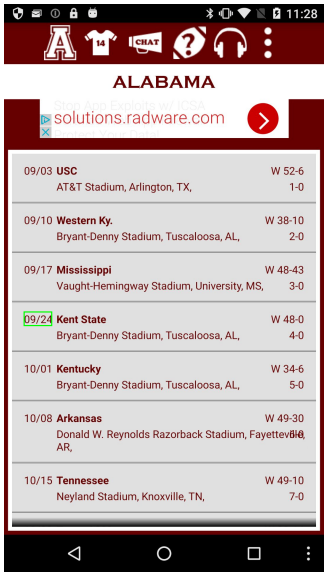


(a) Question annotation UI

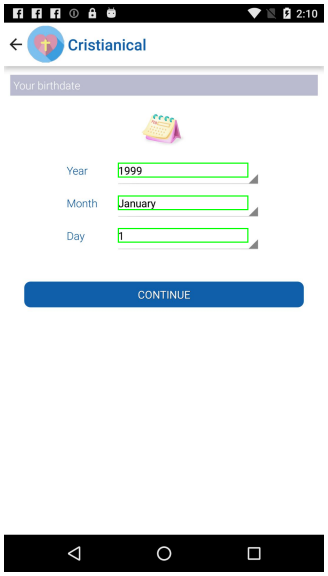


(b) Answer annotation UI

Figure 5: Annotation interfaces.

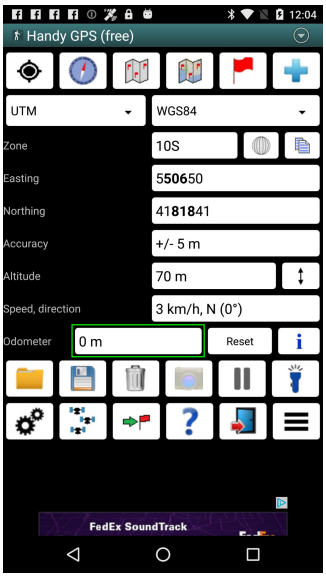


Question	When was the match held at Kent State?
UI elements	• [09/24] • [09/24]
Full answers	• The match was held at Kent State on September 24. • The match was held on September 24.
Generated Short Ans.	• 09/24 • September 24 • September 24th • 9/24

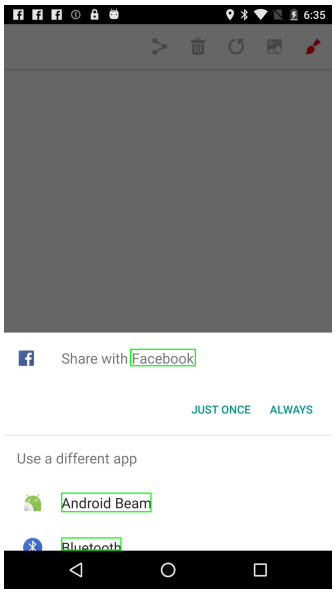


Question	What is the birth date of the user?
UI Elem.	• [1999], [January], [1] • [January], [1], [1999] • [1], [January], [1999]
Long Ans.	• The birth date of the user is January 1, 1999. • The user's birth date is January 1, 1999. • The birth date is January 1, 1999.
Generated Short Ans.	• 1/1/1999 • January 1, 1999 • 1 January 1999 • 1 January, 1999 • January 1st, 1999

Question	What is the status of "Open links inside the app"?
UI Elem.	• [off] • [off]
Long Ans.	• The status of "Open links inside the app" is "off". • The status is "off".
Generated Short Ans.	• off • disabled



Question	What is the odometer reading?
UI Elem.	• [0 m] • [0 m]
Long Ans.	• The odometer reading is 0m. • The odometer reading is 0 m.
Generated Short Ans.	• 0 m • 0 meters



Question	What other applications can be used?
UI Elem.	• [Android Beam], [Bluetooth] • [Facebook], [Android Beam]
Long Ans.	• The applications that can be used are "Facebook", "Android Beam", and "Bluetooth". • The other applications that can be used are "Android Beam" and "Bluetooth".
Generated Short Ans.	• Android Beam, Bluetooth • Android Beam and Bluetooth • "Android Beam", "Bluetooth" • "Android Beam" and "Bluetooth" • Facebook, Android Beam, Bluetooth • Facebook and Android Beam and Bluetooth • Facebook or Android Beam or Bluetooth

Table 9: Examples from ScreenQA dataset

C Dataset Annotation Analysis

Additional dataset annotation analysis is provided in this section. Table 10 shows the remaining question categories and examples continuing from Table 4. Figure 6 shows distribution of question types (e.g., Wh- questions, Yes/No questions, etc.), regardless of the subject.

D Evaluation Configurations

In Section 6 we report the performance of various models on ScreenQA tasks in a zero-shot setting and after fine-tuning. We provide additional details as to how we evaluated the baselines for the corresponding model sizes.

D.1 Zero-shot

Question answering is one of the most common tasks for LLMs and VLMs. Since the output format of the *SQA-S* task is similar to other existing benchmarks, we attempt evaluating publicly available models in zero-shot setting. Specifically, we focus our evaluations on Fuyu-8b[5] (Bavishi et al., 2023), Gemini 1.5 Flash[6], Gemini 1.5 Pro[7] (Gemini Team Google, 2023), and GPT-4o[8] (OpenAI et al., 2024). We refer to Table 5 for the results.

Further, we describe for each model the prompt used in the evaluation is an instruction followed by a question. Fuyu-8B model came with an author recommendation for a prompt, which we made use of [9]:

```
Answer the following DocVQA question based on the
image. \n
```

For GPT-4o we optimized the prompt using 10 examples from the validation split. The one we picked was:

```
Answer the question based on the screenshot only. Do
 not use any other sources of information. The
answer should be succinct and as short as possible.
If the answer is a text from the image, provide it
exactly without rephrasing or augmenting. If there
is no answer on the image, output "<no answer>".\n
```

The same prompt was re-used for the Gemini model evaluation. Although this may represent a disadvantage for Gemini compared to GPT-4o, our results indicate that Gemini 1.5 Flash is on par with and 1.5 Pro outperforms GPT-4o using it.

D.2 Fine-tuning

The training data for SQA-L, SQA-UIC and SQA-UIC-BB tasks has one annotation per example (with a few exceptions), so it's usage for fine-tuning is straightforward for all models. Short answers, on the other hand, were generated automatically (see Section 4.5), resulting in multiple answers per question for the SQA-S task. The fine-tuning setup for PaliGemma 3B (Beyer et al., 2024) and ScreenAI 5B (Baechler et al., 2024) models randomly select one answer each time. Through sufficient epoch, it is however likely that all answer variants are utilized.

For Gemini 1.5 Flash fine-tuning on SQA-S task, however, one answer per question was randomly selected before training, limiting the diversity of answers observed by the model. We therefore ran fine-tuning twice for 2 different random selections, and reported the best obtained results among the two experiments.

E Observed prediction errors on SQA-S task

While Section 6 contains evaluations of different models in different settings to provide baselines, in this paper we deliberately refrain from making conclusions about upsides and downsides of each individual model. Instead, in Table 11 we provide examples from SQA-S task of some errors models made during evaluation, as a demonstration of complexity of this dataset. While this may not be a complete list, we categorized those errors into:

- *Misinterpreted question*: the predicted answer answers a different question.
- *Misinterpreted screenshot info*: the predicted answer is based on the screenshot information that has a different meaning.
- *Hallucination*: the predicted answer is not based on the screen at all.
- *Modified or misread content*: the predicted answer originates from correct answer but contains typos or is incorrectly transformed into something different.
- *Lack of understanding/reasoning to identify/-compose answer*: the predicted answer suggests poor model's screen understanding or reasoning capabilities.

[5] Fuyu-8b [6] Gemini 1.5 Flash [7] Gemini 1.5 Pro
[8] GPT-4o [9] See Fuyu-8B discussion

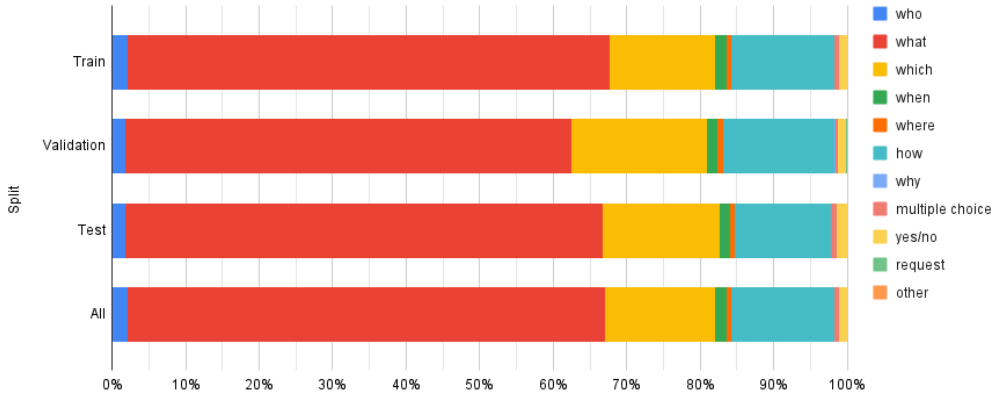
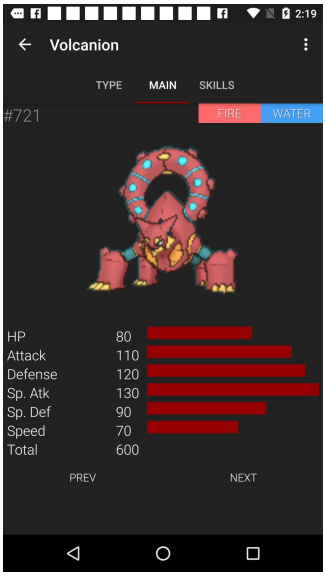


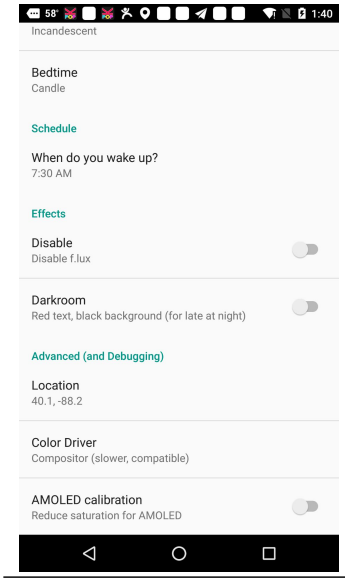
Figure 6: Histogram for the types of questions.

Category	%	Examples	
Subtotal from Table 4	83.8		
Signup/login	1.6	Which application can be used to sign up / login?	What are the alternative choices for signing up?
Version information	1.6	What is the version number?	What is the new feature in version v3.1.3?
Weather	1.5	What is the range of temperature shown on Sunday?	What is the weather forecast for Sunday?
Score & value	1.4	What is height/weight of the person?	What is the score?
Yes/No	1.1	Is there any travel plans?	Is there any favorite?
Phone number	1.0	What is the phone number?	What is the prefix for the international mobile number?
# of Stars	0.8	What is the star rating?	How many stars are given to the product?
Share/sharing	0.8	Which application can be used to share?	Where can I share this application?
Age	0.8	How old is ...?	What is the age?
Percentage	0.7	What is the percentage of ... ?	What is the brightness percentage for foreground?
Settings	0.6	What is the setting of ... ?	Which settings are switched on?
Quantity amount	0.6	How much fat is there?	What is the amount?
Permission	0.5	Which application is asking for permissions?	What permissions are required for MyCarTracks?
# of Likes	0.5	How many likes for ... ?	How many likes does ... get?
Country	0.5	What is the name of the country?	Which country has the +54 code?
Distance	0.5	What is the visibility distance?	How far is it from ... ?
# of Reviews	0.4	What is the number of comments on ... ?	How many comments?
Website	0.3	What is the url?	What's the website address?
Gender	0.3	What is the gender?	Which gender is displayed on the screen?
How to	0.3	How to start on boot?	How to pronounce his name?
Currency	0.3	What is the currency?	What is the currency for the price?
Unit of measurement	0.2	What is the unit of temperature?	What is the unit of weight and length?
Language	0.1	Which language is used in the setting?	Which language is being translated into which language?
Color	0.0	What is the UI color?	What is the amount of green color?
Total	100.0		

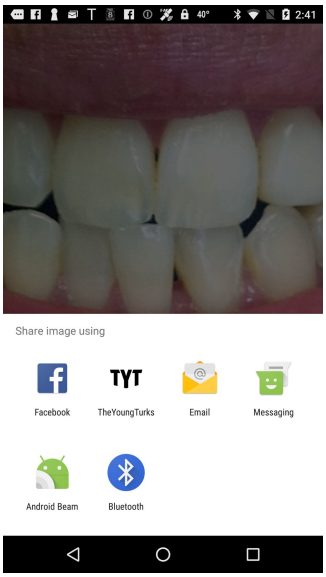
Table 10: Remaining question categories and examples (*cont.* from Table 4)



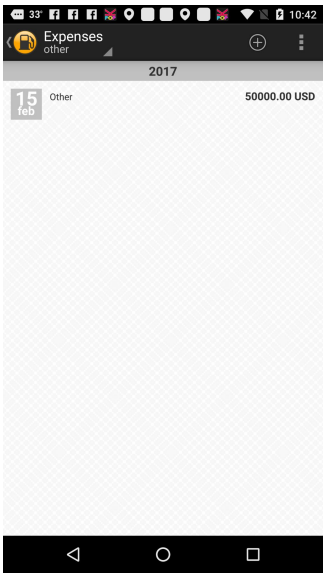
Question	What is the total number?
Correct answer	600
Incorrect answer	721
Error category	Misinterpreted question
Explanation	The serial number present on the screen was incorrectly identified as the total.



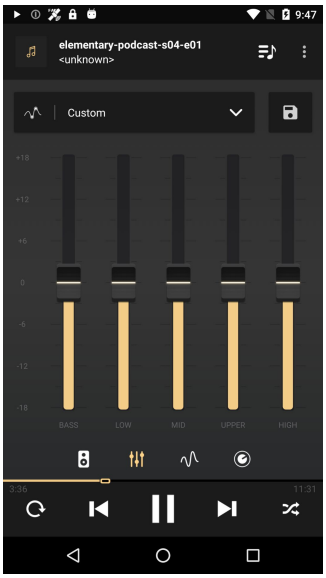
Question	For which setting is the "Compositor" option selected?
Correct answer	"Color Driver" setting
Incorrect answer	Advanced (and Debugging)
Error category	Misinterpreted question
Explanation	"Advanced (and Debugging)" is a group of settings, and "Color Driver" is a single setting in that group, which is the correct answer.



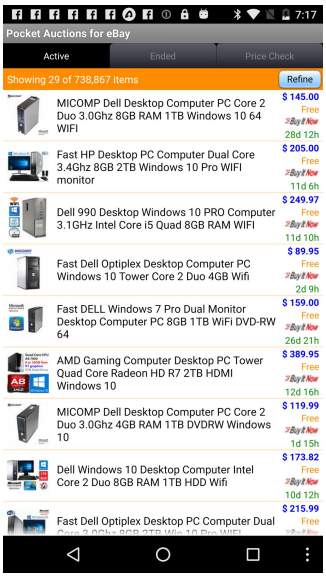
Question	Through which applications can we open the browser?
Correct answer	<no answer>
Incorrect answer	Facebook, TheYoungTurks, Email, Messaging, Android Beam and Bluetooth
Error category	Misinterpreted question
Explanation	Predicted the ways to share content, instead of opening the browser.



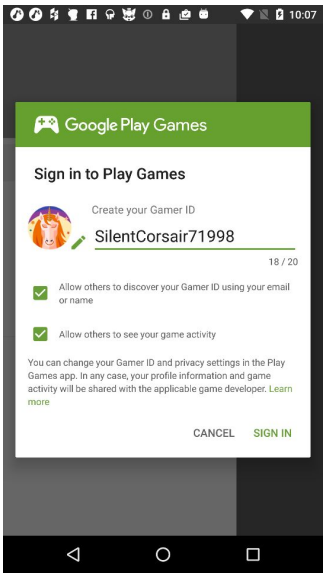
Question	What's the title of the expense amount for February 15th?
Correct answer	Other
Incorrect answer	50000.00 USD
Error category	Misinterpreted screenshot info
Explanation	The expense amount of 50000.0 USD was mistakenly identified as the title.



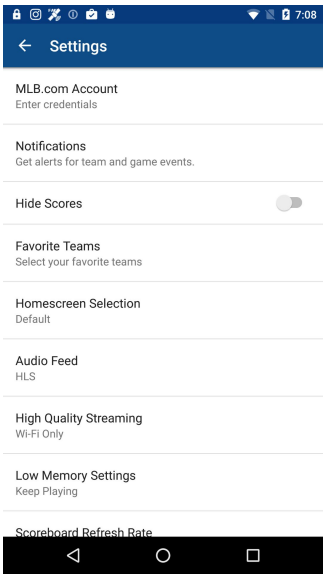
Question	How long is the audio?
Correct answer	11 minutes and 31 seconds
Incorrect answer	3:36
Error category	Misinterpreted screenshot info
Explanation	The current audio playing position at 3:36 was identified as the total duration instead of 11:31.



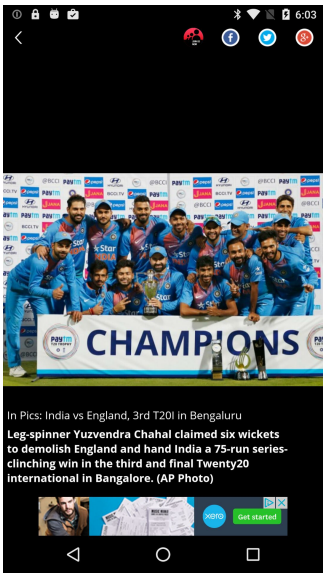
Question	What is the number of shown items?
Correct answer	29
Incorrect answer	738,867
Error category	Misinterpreted screenshot info
Explanation	The total number of items is 738,867, while 29 is the shown number.



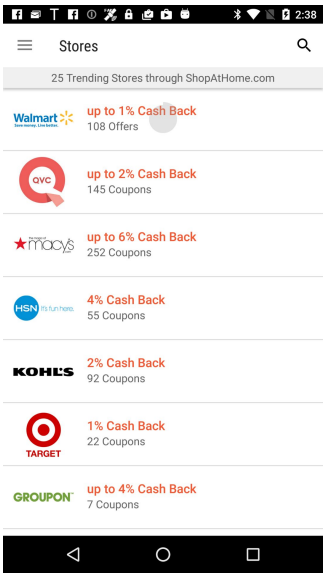
Question	When was the article published?
Correct answer	<no answer>
Incorrect answer	18 / 20
Error category	Misinterpreted screenshot info
Explanation	18 / 20 represents the number of characters in a text field and not the date.



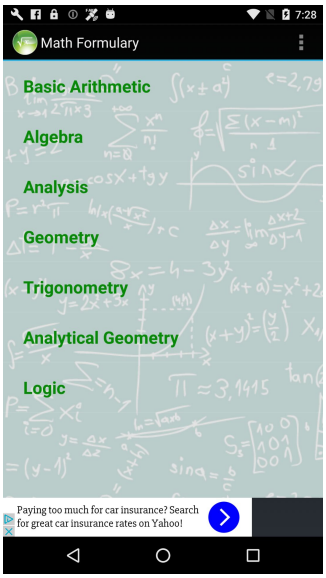
Question	Which option is selected for high quality streaming?
Correct answer	Wi-Fi Only
Incorrect answer	"Only for VIPs" option
Error category	Hallucination
Explanation	"High Quality Streaming" setting has value "Wi-Fi Only".



Question	Who has taken six wickets?
Correct answer	Yuzvendra Chahal
Incorrect answer	Lelebhim Yuzvendra Chahal
Error category	Hallucination
Explanation	Possibly the word "Leg-spinner" was misread or hallucinated into "Lelebhim".

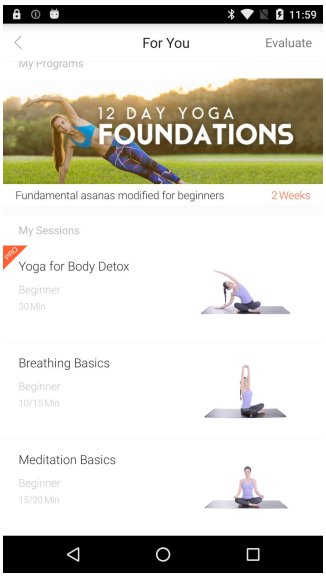


Question	Which company is giving 252 coupons?
Correct answer	macy's
Incorrect answer	"mady's"
Error category	Modified or misread content
Explanation	"Macy's" was misspelled.

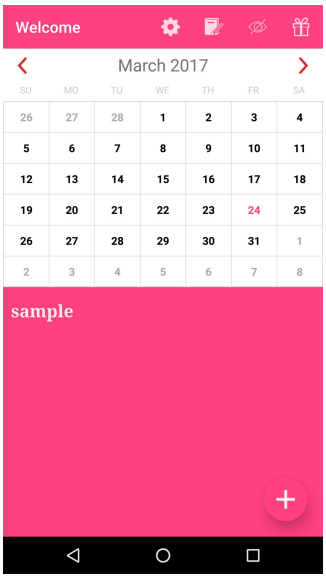


Question	What is the application name?
Correct answer	Math Formulary
Incorrect answer	Math Formulas Free
Error category	Modified or misread content
Explanation	Hallucination of "Forumlas Free" instead of "Formulary"

Question	What are the different types of genres?
Correct answer	"AVANT-GARDE", "INTERNATIONAL", "BLUES", "JAZZ", "CLASSICAL" and "NOVELTY"
Incorrect answer	"AVANT-garde", "INTERNATIONAL", "BLUES", "JAZZ", "CLASSICAL" and "SAVELY"
Error category	Modified or misread content
Explanation	Hallucination of "SAVELY" instead of "NOVELTY"



Question	How many minutes does a beginner have to do the breathing basics session?
Correct answer	10 to 15 minutes
Incorrect answer	10 minutes and 15 minutes
Error category	Lack of understanding/reasoning to identify/-compose answer
Explanation	The UI indicates that the breathing session is "10/15 min", which from the context should be interpreted as 10 to 15 mins, rather than 10 and 15 mins, separately.



Question	What is the day on the 17th of March?
Correct answer	FR
Incorrect answer	TH
Error category	Lack of understanding/reasoning to identify/-compose answer
Explanation	The day of the week was not correctly identified from the calendar.

Table 11: Examples of errors made by the models from ScreenQA dataset on SQA-S task.